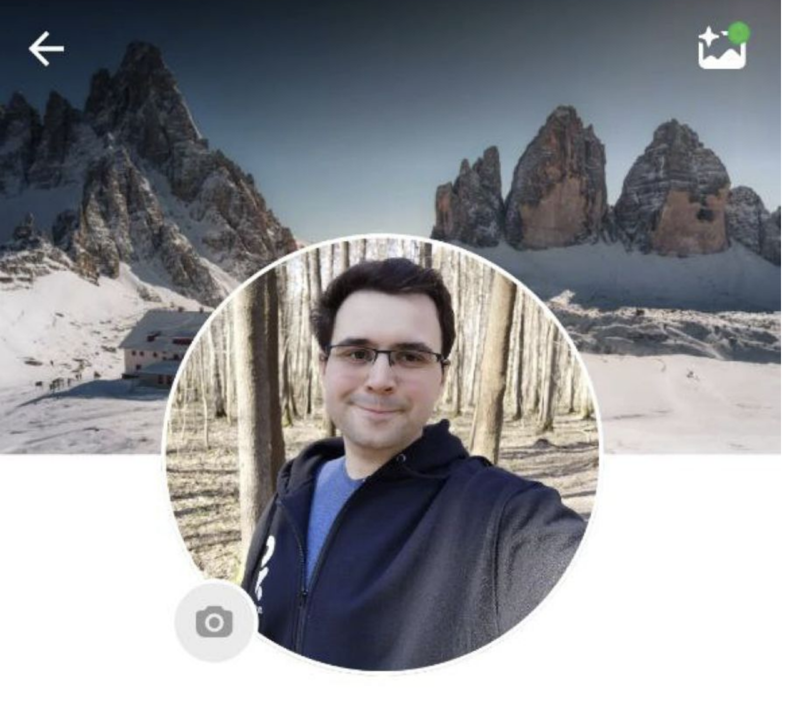
Машинное обучение в продакшне — это просто! Нужно только….

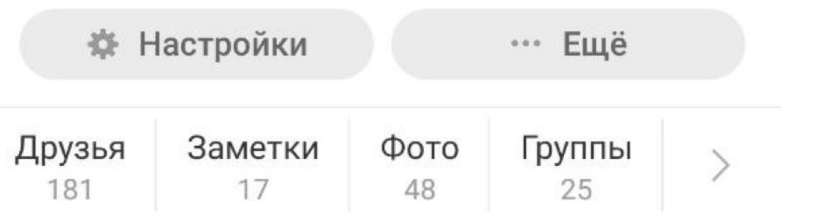
Михаил Марюфич, Одноклассники











Инженер по машинному обучению

Разрабатываю ML-решения
Выкатываю их в прод
Улучшаю ML-инфраструктуру

Одноклассники



- Около 41 млн жителей России ежемесячно*
- 168 миллионов реакций в день

*По данным MediaKit, 2021

Одноклассники

>500 сервисов
>9000 машин
6 дата-центров



>80 TB RAM
>100 PB Storage
>20K Cores
>24 TB новых данных ежедневно

Машинное обучение — это важно!



- Рекомендации постов, групп, музыки, друзей и т.д.
- Защита от спама
- ...



Более 100 моделей



Как может выглядеть работоспособный продакшн, если пойти по пути **наименьшего сопротивления,** и почему это не работает, когда **много моделей** и **ответственность** перед пользователями!



Рекомендации



Сервис, где мы показываем пользователю наиболее интересные посты среди всего контента Одноклассников*

Актуальное Коронавирус Новости Рядом Сад и огород Культура Рецепты Ещё
Поиск по темам

- Не подскажете, который час? - Без пяти минут май...

ТЁПЛОЕ НАСТРОЕНИЕ

Швейцария

Пасмурный день в долине водопадов...

Таинственные дали

Баланс

Гармония Природы

Самые красивые места Земли

* От групп и авторов, на которых пользователь еще не подписан

Рекомендации



Сервис, где мы показываем пользователю наиболее интересные посты среди всего контента Одноклассников*

МЛ в рекомендациях

- **найти** наиболее интересные пользователю **источники**, среди 11+ млн групп и авторов

* От групп и авторов, на которых пользователь еще не подписан

Актуальное Коронавирус Новости Рядом Сад и огород Культура Рецепты Ещё

Поиск по темам

- Не подскажете, который час? - Без пяти минут май...

ТЁПЛОЕ НАСТРОЕНИЕ

Швейцария

Пасмурный день в долине водопадов...

Таинственные дали

Баланс

Гармония Природы

Самые красивые места Земли

Как выглядит продакшн



100 мс

по **userId** достаем
список источников

User

ok.ru/discovery

ok-feeds-recommender

storage

Storage/Feature store



Умеет быстро втягивать данные из **kafka** и отдавать данные по ключу в онлайне.



Мы используем его для хранения и быстрого извлечения по id различной информации, такой как **список рекомендуемых групп, вектор интересов** и др.

Как выглядит продакшн



Все, что нам нужно сделать — это сделать так, что в хранилище по userId будут записаны нужные нам данные.

по **userId** достаем
список источников

100 мс

ok.ru/discovery

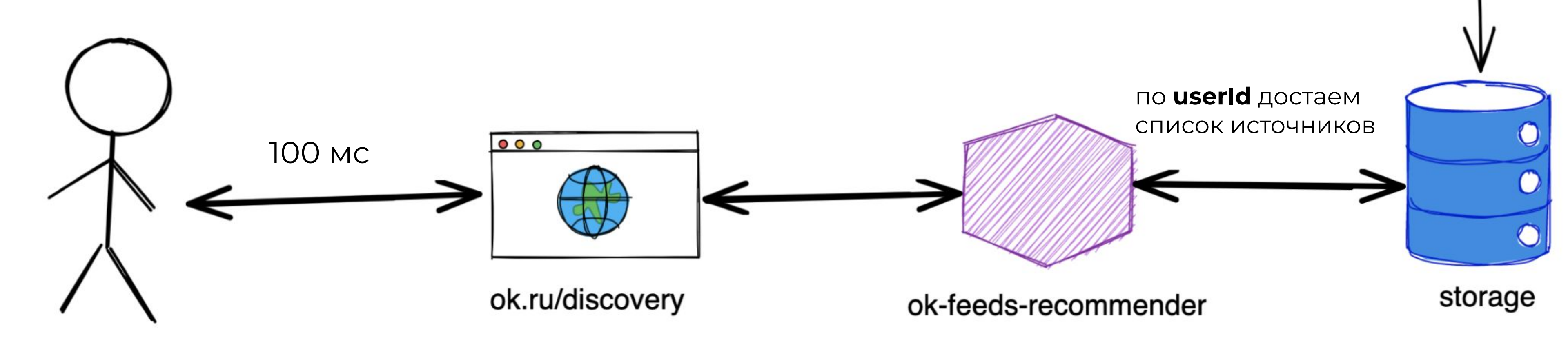
ok-feeds-recommender

storage

User

Как выглядит продакшн

Все, что нам нужно сделать — это сделать так, что в хранилище по userId будут записаны нужные нам данные.



Как это происходит?



Jira

Улучшить систему рекомендаций

MLE

Собрать данные → Обучить модель → Провалидировать модель → Задеплоить модель

Как это происходит?



Jira

Улучшить систему рекомендаций

MLE

Собрать данные → Обучить модель → Провалидировать модель → Задеплоить модель



Как это происходит?



Jira

Улучшить систему рекомендаций

MLE

Собрать данные → Обучить модель → Провалидировать модель → Задеплоить модель

Spark

Как это происходит?



Jira

Улучшить систему рекомендаций

MLE

Собрать данные → Обучить модель → Провалидировать модель → Заделплоить модель

Spark

Как это происходит?



Jira

Улучшить систему рекомендаций

MLE

Собрать данные → Обучить модель → Провалидировать модель → Задеплоить модель

Spark

jupyter

Как это происходит?



Jira

Улучшить систему рекомендаций

MLE

Собрать данные → Обучить модель → Провалидировать модель → Задеплоить модель

Spark

jupyter

?



Наша цель — это не задеплоить модель, а сделать пользователей счастливыми как можно быстрее!



Путь к счастью пользователя



Разрабатываем хорошую модель

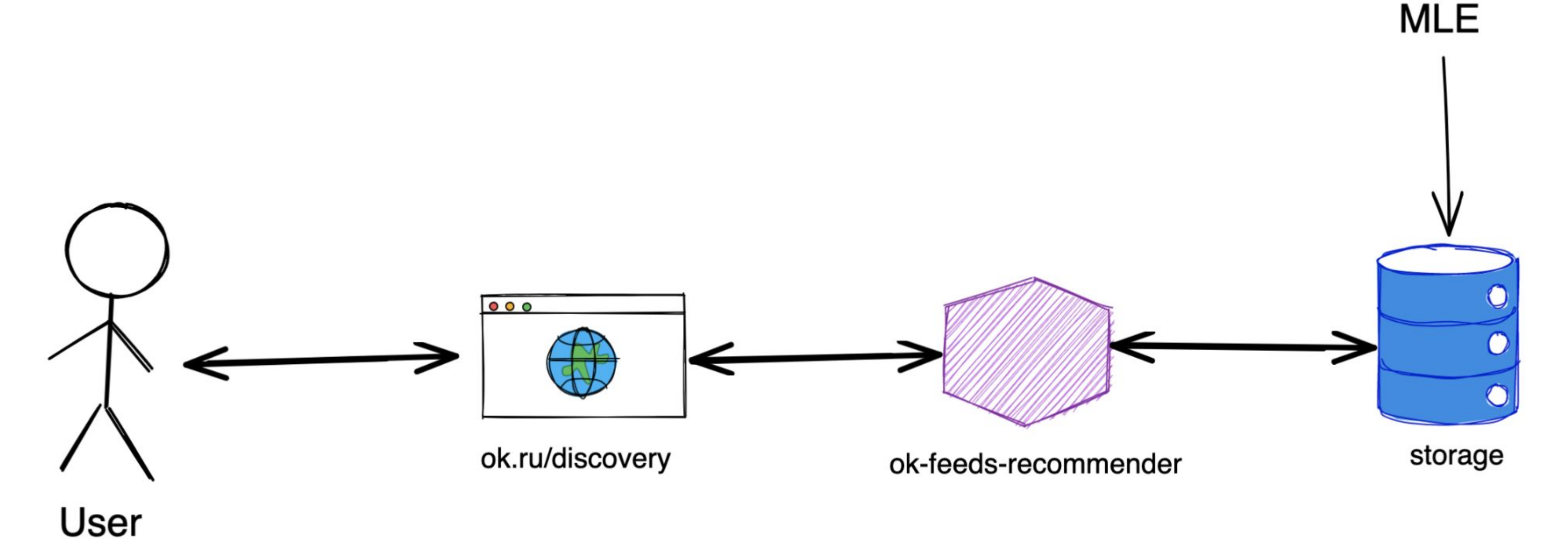
Она выдает классные рекомендации

Пользователь счастлив



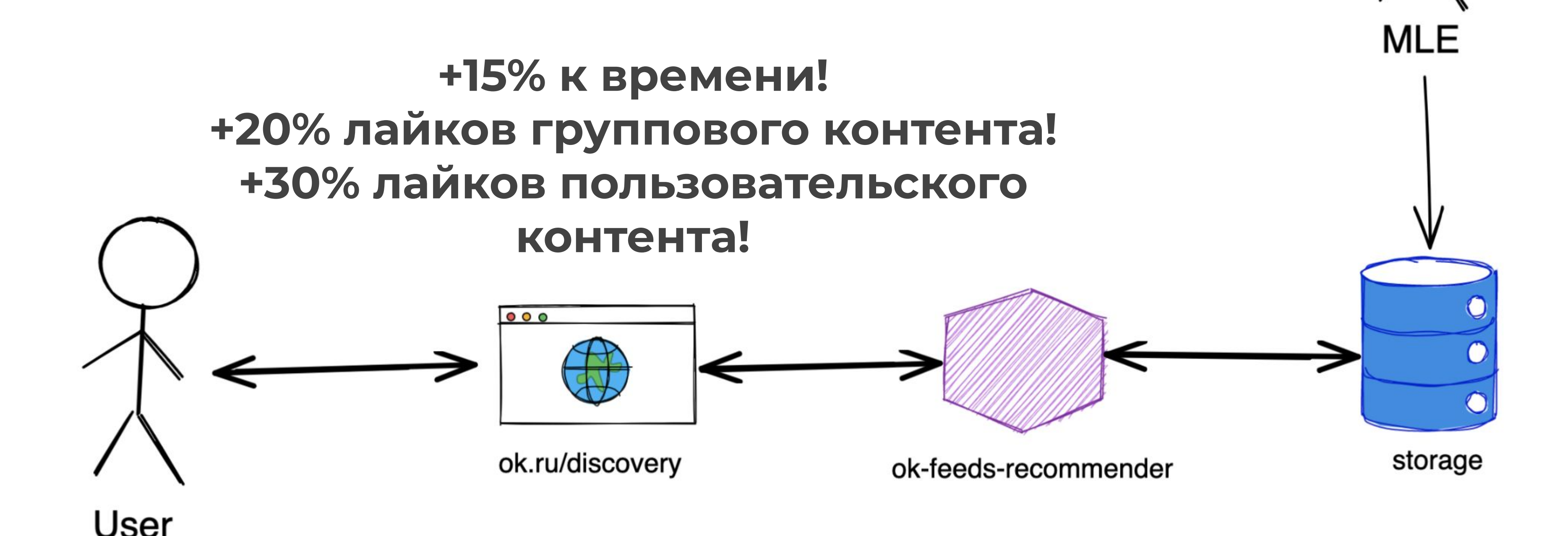
Проверка гипотезы

Для **четверти** пользователей заливаем предсказания модели руками и проводим A/B-тест



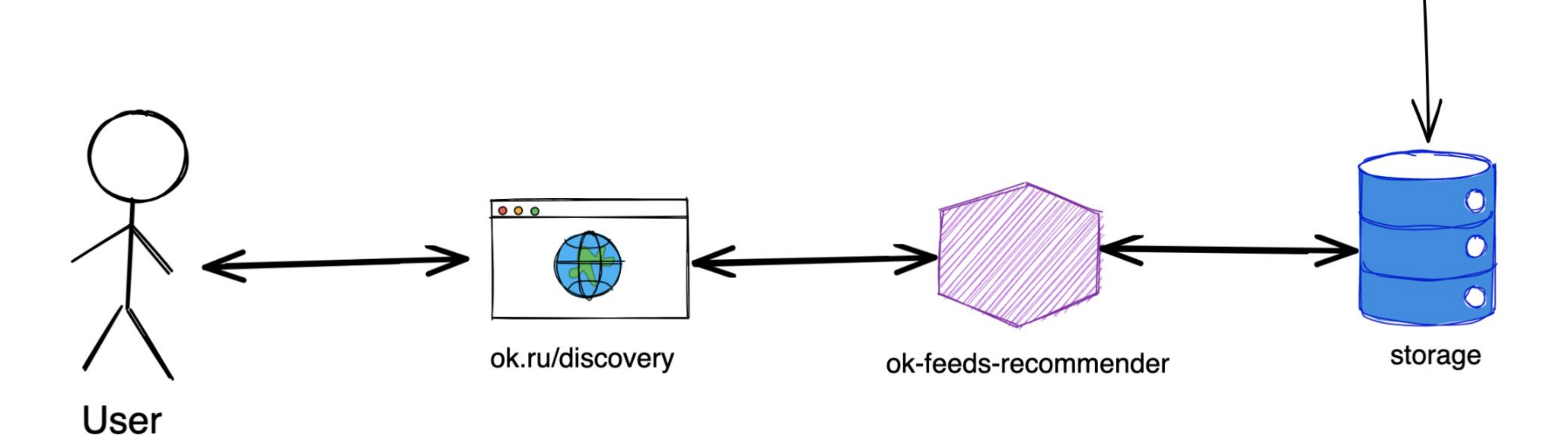
Проверка гипотезы

Для **четверти** пользователей заливаем предсказания модели руками и проводим A/B-тест



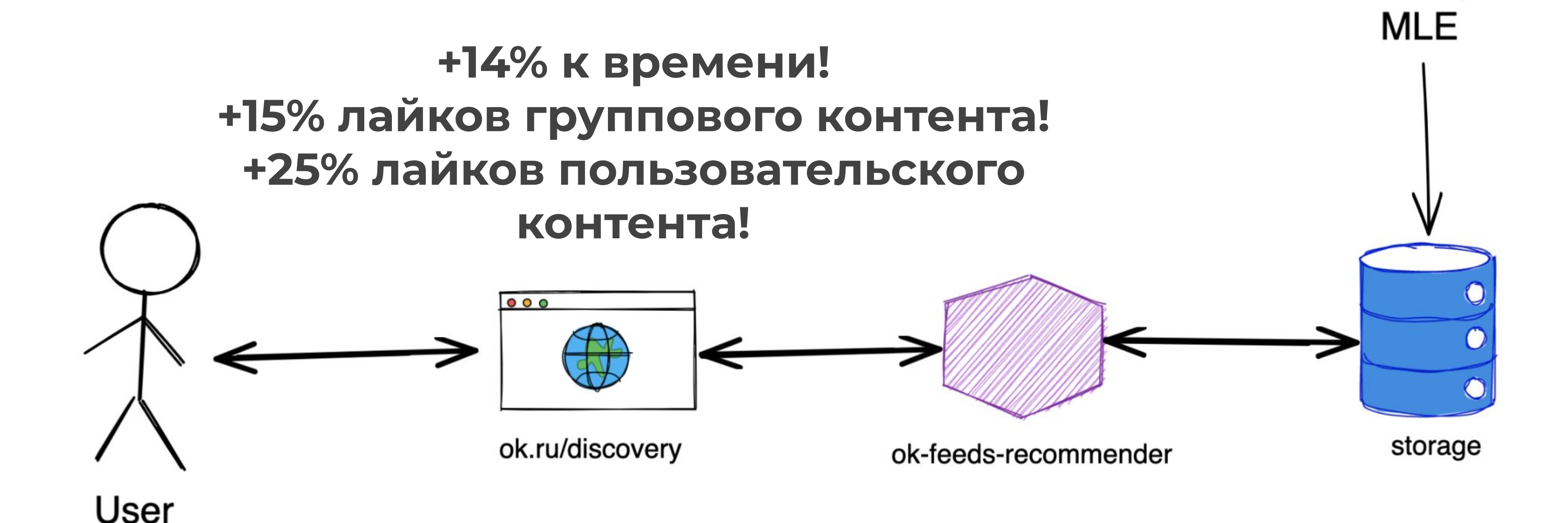
Раскатка на всех

Для **всех** пользователей заливаем предсказания модели руками



Раскатка на всех

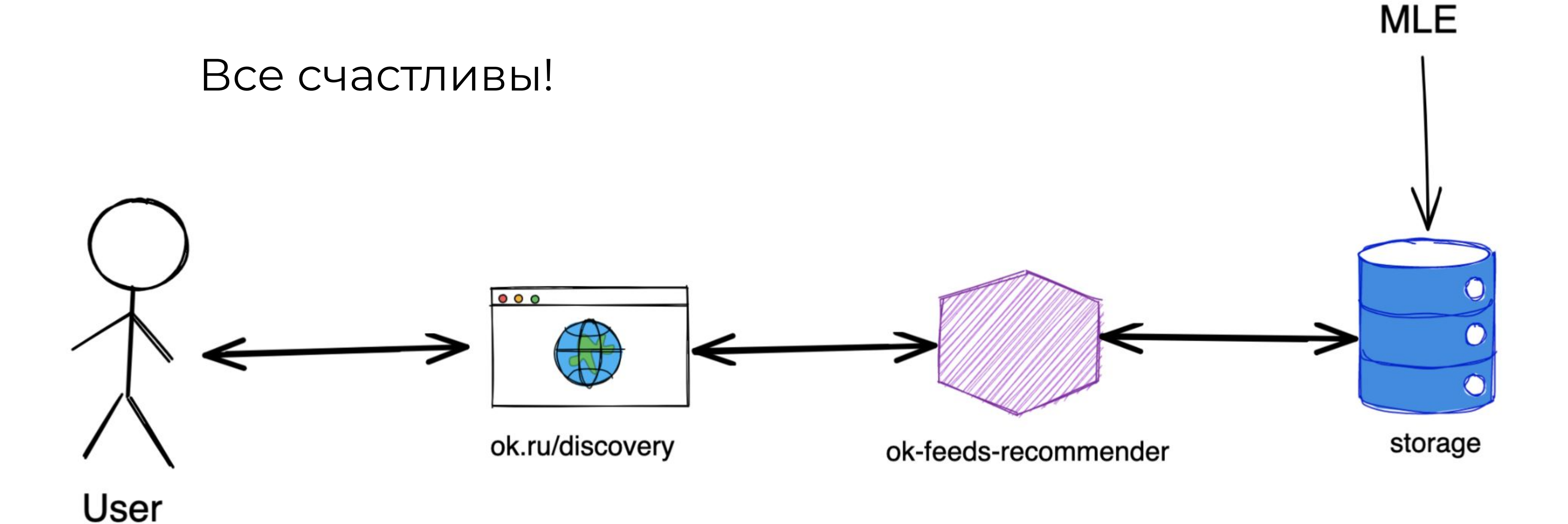
Для **всех** пользователей заливаем предсказания модели руками



Раскатка на всех

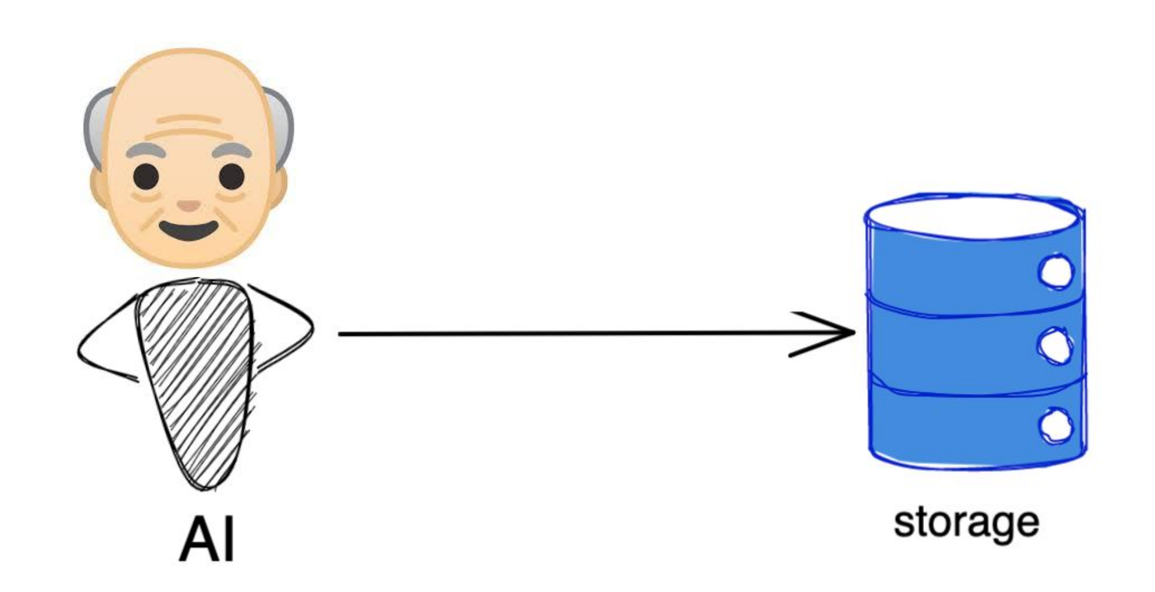
Для **всех** пользователей заливаем предсказания модели руками

Все счастливы!



Наблюдение: рекомендации устаревают



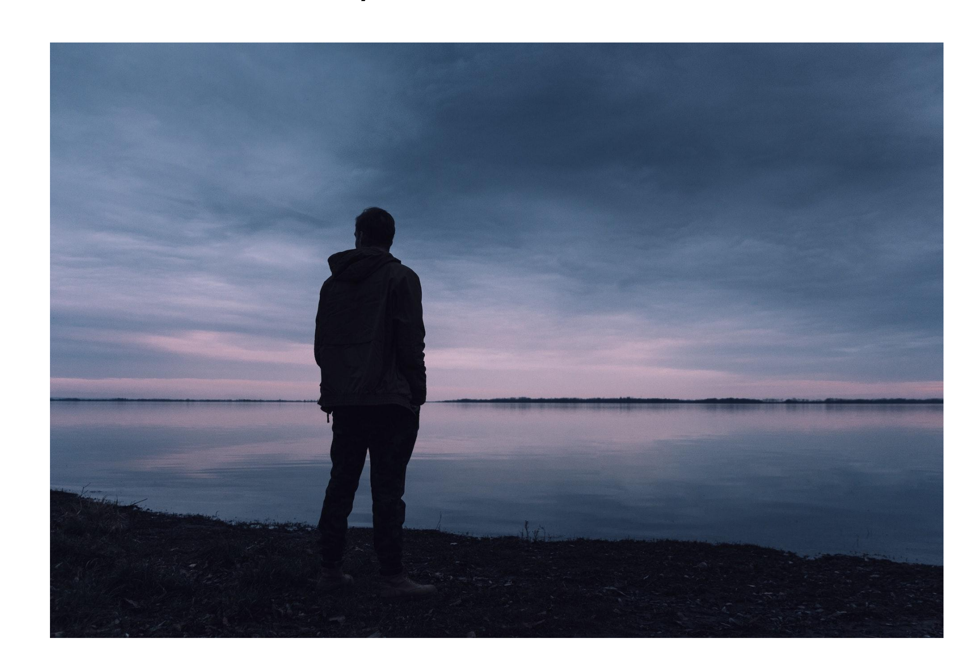


Рекомендации нужно обновлять раз в неделю



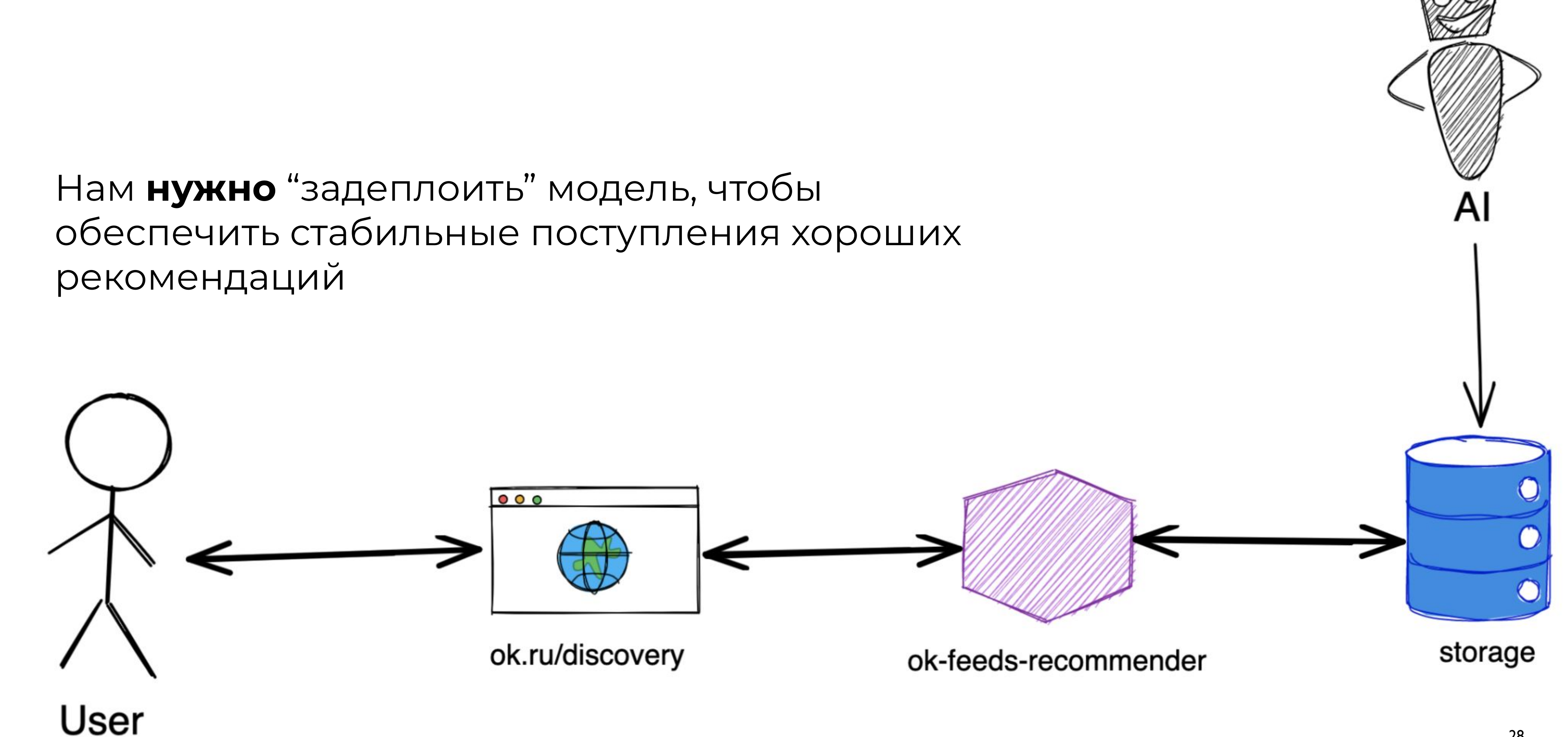
Будет работать, НО для поддержания работы продакшна потребуются ручные действия (раз в неделю)

Когда много моделей очень сильно возрастает вероятность факапа.



Нам нужно деплоить модель

Нам **нужно** “задеплоить” модель, чтобы обеспечить стабильные поступления хороших рекомендаций



Как работает модель?



пользовательская активность

активность друзей

подписки пользователя

признаки пользователя (кол-во друзей, возраст, пол, геолокация)

признаки источников (кол-во подписчиков, тематика)

Нейронка

вектор пользовательских интересов

[Как я перестал бояться и научился любить нейронные сети]

Как работает модель?



Вектор пользовательских интересов

0.2

0.5

0.9

Вектор источника 1

Вектор источника 2

Вектор источника 3

Как работает модель?



пользовательская активность

активность друзей

подписки пользователя

признаки пользователя (кол-во друзей, возраст, пол, геолокация)

признаки источников (кол-во подписчиков, тематика)

Нейронка

вектор пользовательских интересов

NVIDIA

Как выглядит ручной процесс



feed/prod_discovery

Считываем данные и готовим фичи

```
val dataset1 = loadDataset("10 days ago:today", "DATASET1_PATH")
val dataset2 = loadDataset("9 days ago:today", "DATASET2_PATH")
val dataset3 = loadDataset("14 days ago:today", "DATASET3_PATH")

val features = prepareFeatures(dataset1, dataset2, dataset3)
saveFeatures(features, "FEATURES_OUTPUT")
```

Считаем пользовательские векторы

```
// ssh srvk4353
// cd /home/user_userovich/my_useful_scripts/prod_discovery/predict_neironka_v32
// sh do_magic.sh
```

Заливаем предсказания в STORAGE

```
val userEmbeddings = spark.read.parquet("USER_EMBEDDINGS")
val sourceEmbeddings = spark.read.parquet("SOURCE_EMBEDDINGS")
val predicts = postprocess(userEmbeddings, sourceEmbeddings)
uploadPredicts(predicts, "PROD_DISCOVERY_TOPIC")
```

Как выглядит ручной процесс



feed/prod_discovery

Считываем данные и готовим фичи

```
val dataset1 = loadDataset("10 days ago:today", "DATASET1_PATH")
val dataset2 = loadDataset("9 days ago:today", "DATASET2_PATH")
val dataset3 = loadDataset("14 days ago:today", "DATASET3_PATH")

val features = prepareFeatures(dataset1, dataset2, dataset3)
saveFeatures(features, "FEATURES_OUTPUT")
```

Считаем пользовательские векторы

```
// ssh srvk4353
// cd /home/user_userovich/my_useful_scripts/prod_discovery/predict_neironka_v32
// sh do_magic.sh
```

Заливаем предсказания в STORAGE

```
val userEmbeddings = spark.read.parquet("USER_EMBEDDINGS")
val sourceEmbeddings = spark.read.parquet("SOURCE_EMBEDDINGS")
val predicts = postprocess(userEmbeddings, sourceEmbeddings)
uploadPredicts(predicts, "PROD_DISCOVERY_TOPIC")
```

Не под контролем GIT
Не было ревью
НЕСТАБИЛЬНО

Inference Pipeline



Inference Pipeline



hadoop HDFS | APACHE Spark | Оркестратор контейнеров

Датасеты с данными для нейронки → Подготовить данные для нейронки → Получить векторы интересов, используя нейронку → Получить итоговые предсказания

Вектора источников → Получить итоговые предсказания

Файл с моделью → Получить итоговые предсказания

Получить итоговые предсказания → Залить предсказания в storage

TensorRT

Apache Airflow

Apache Airflow



Один из самых популярных оркестраторов данных.

- отлично подходит для batch-сценариев
- встроенная работа с расписаниями
- позволяет выстраивать сложные зависимости
- удобен для работы в гетерогенной среде



Текущая схема



Jira

Улучшить систему рекомендаций

MLE

Apache Airflow

Обеспечить inference модели

Собрать данные → Обучить модель → Провалидировать модель → Поместить модель в место, доступное для прода

Spark

jupyter

hadoop HDFS

Выводы



- На продакшн **не должен** влиять человеческий фактор.
- Для экспериментов такой подход имеет место быть.

Организационные решения



- Без автоматизации раскатываем максимум на четверть аудитории для временного А/Б-эксперимента.

- Стремимся вводить автоматизацию на этапе первых экспериментов (автоматизировано использование моделей для 90% от всех экспериментов).

Мы хотим стабильности



Задача: понимать, если состояние нашей системы деградирует

Мониторинг сервисов



1) Время отдачи данных со стороны storage
2) Время построения рекомендаций
3) Количество запросов
4) Количество ошибок

Продуктовые метрики



1) Количество лайков в разрезе разных типов контента
2) Количество комментариев
3) Количество запросов в друзья
4) Время проведенное на сайте
5) ...

Мониторинг inference pipeline



Контролируем, что каждая из частей пайплайна отрабатывает

Airflow · DAGs · Security · Browse · Admin · Docs · 21:11 UTC · RH

DAGs

All 26 · Active 10 · Paused 16 · Filter DAGs by tag · Search DAGs

DAG	Owner	Runs	Schedule	Last Run	Recent Tasks	Actions	Links
example_bash_operator (example, example2)	airflow	2	0 0 * * *	2020-10-26, 21:08:11	6		
example_branch_dop_operator_v3 (example)	airflow		*/1 * * * *				
example_branch_operator (example, example2)	airflow	1	@daily	2020-10-23, 14:09:17	11		
example_complex (example, example2, example3)	airflow	1 1	None	2020-10-26, 21:08:04	37 37		
example_external_task_marker_child	airflow	1	None	2020-10-26, 21:07:33	2		
example_external_task_marker_parent	airflow	1	None	2020-10-26, 21:08:34	1		
example_kubernetes_executor (example, example2)	airflow		None				
example_kubernetes_executor_config (example3)	airflow	1	None	2020-10-26, 21:07:40	5		
example_nested_branch_dag (example)	airflow	1	@daily	2020-10-26, 21:07:37	9		
example_passing_params_via_test_command (example)	airflow		*/1 * * * *				

История про инцидент



Резко упало количество классов и комментариев

История про инцидент



Резко упало количество классов и комментариев

В итоге нашли, что поплыли значения фичей

Проблема



При внешней стабильности системы — могут выкатиться некачественные рекомендации.

мониторинг сервисов

мониторинг ML

Мониторинг данных



Следим за изменением статистик данных на вход и выход, допустимыми значениями фичей и предсказания.

Среднее значение Feature_1

Как это работает



Подготовить данные для нейронки

Проверка данных

success

Получить векторы интересов, используя нейронку

fail

Выслать alert







Считаем метрики машинного обучения в онлайне (насколько точно мы предсказываем источники по постам, которые кликают/лайкают и т.д.)

Точность предсказаний

80
60
40
20
0
6 мая
7 мая
8 мая
9 мая

Результаты



- Поверили в то, что мониторинг данных действительно важен

- Предотвращено некоторое количество похожих, тяжело расследуемых, инцидентов.

Текущая схема



Jira

Улучшить систему рекомендаций

MLE

Apache Airflow

Обеспечить inference модели

Собрать данные → Обучить модель → Провалидировать модель → Поместить модель в место, доступное для прода

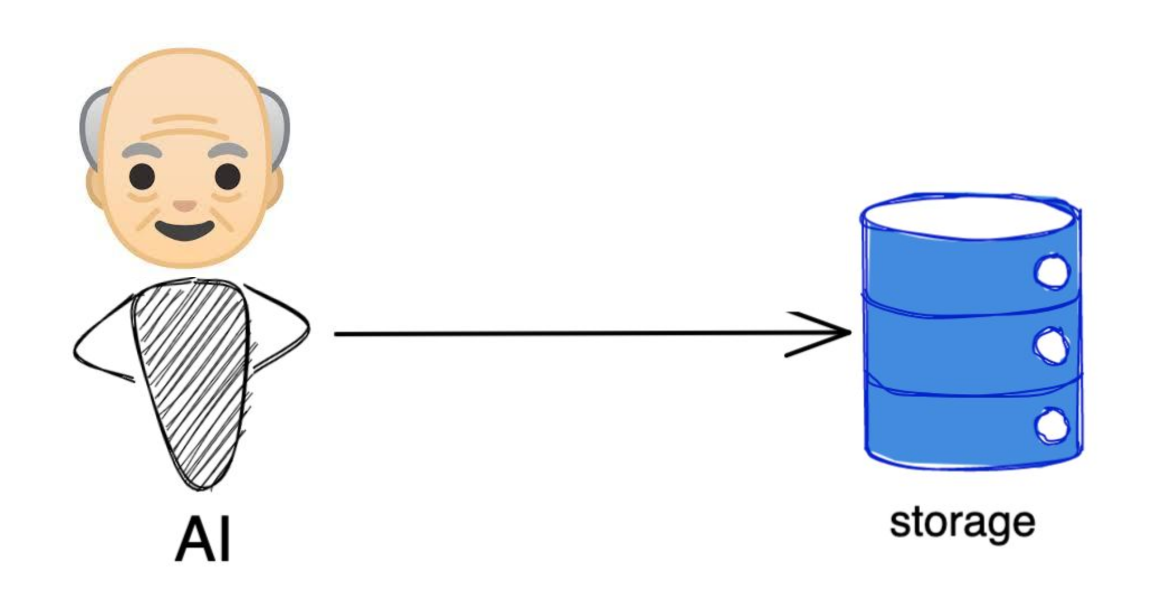
Spark

jupyter

hadoop HDFS

Наблюдение 1: ML-модели — “устаревают”!





“Устаревшую” ML-модель следует обновить

Наблюдение 2: Мы не знаем, что он делает!



Jira

Улучшить системы рекомендаций

MLE

Apache Airflow

Обеспечить inference модели

BLACKBOX

Поместить модель в место, доступное для прода

hadoop HDFS

Почему модели устаревают?



Изменяющийся **мир**



- Меняется пользовательское поведение
- Появляются новые источники

Почему модели устаревают?



Изменяющийся **мир**

- Меняется пользовательское поведение
- Появляются новые источники

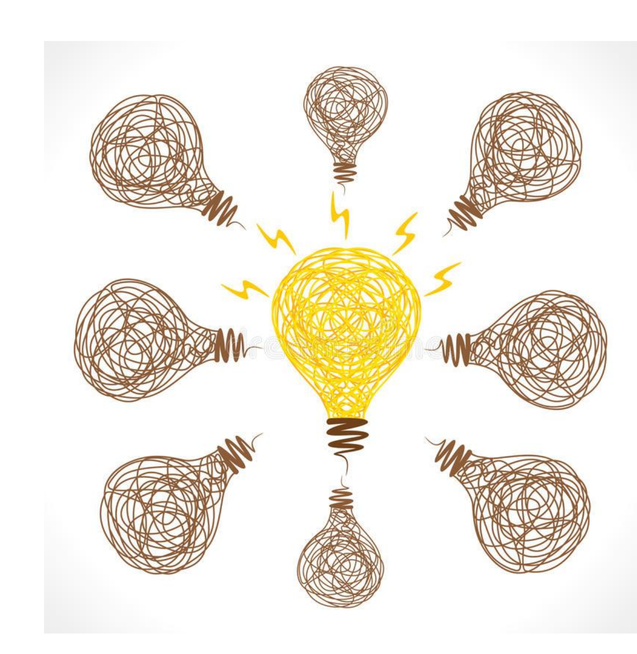


Нужно уметь переобучать модель на новых данных

Почему модели устаревают?



Появление новых идей (ведущих к увеличению счастья пользователей):



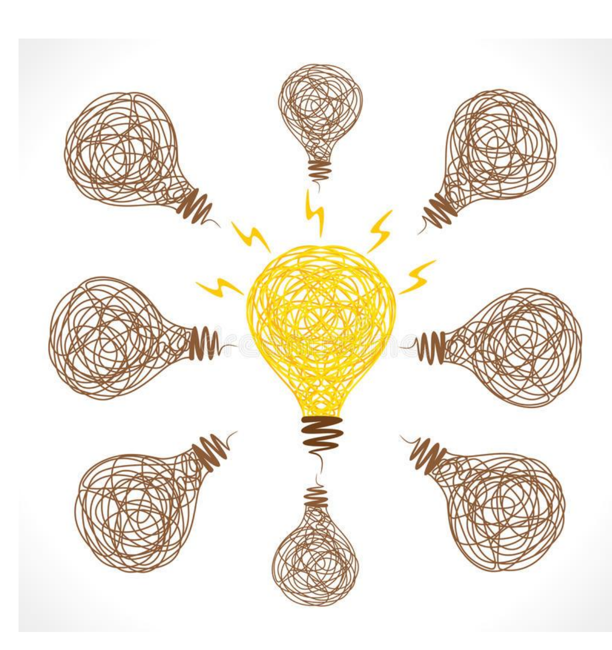
- Придумал новый признак
- Придумал новый способ собрать датасет
- Вышла новая статья, нужно срочно реализовать

Почему модели устаревают?



Появление новых идей (ведущих к увеличению счастья пользователей):

- Придумал новый признак
- Придумал новый способ собрать датасет
- Вышла новая статья, нужно срочно реализовать



Нужно уметь эффективно экспериментировать




Для продакшна **критично:**
1) умение переобучать модель на новых данных
2) умение эффективно экспериментировать

Проблемы в текущем процессе

1) Ручной
2) Непрозрачный процесс экспериментирования
3) Нет способа воспроизвести продакшн-модель, а следовательно, и надежно внести изменение





Корень проблемы





DS делают **модели**



DS делают **пайплайны**

Почему сложно?





Почему сложно?



На каждом из этапов — фиксируем слишком мало информации

MLE

Улучшить систему рекомендаций

Собрать данные

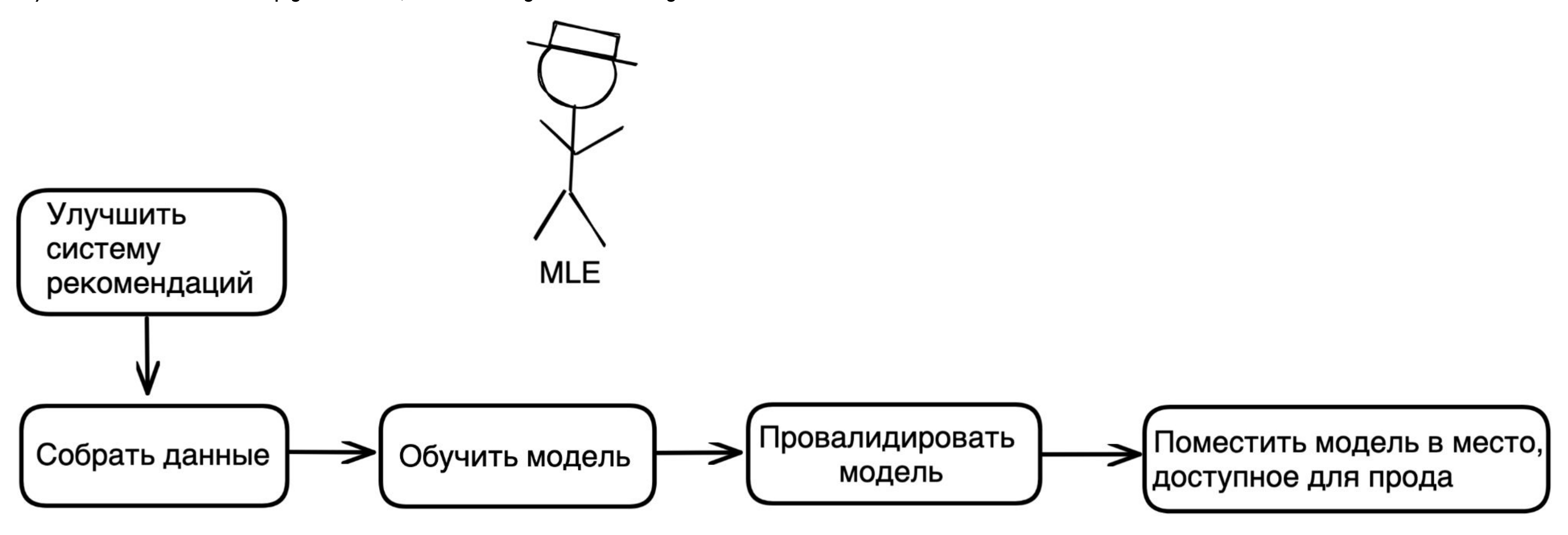
Обучить модель

Провалидировать модель

Поместить модель в место, доступное для прода

Очевидное решение

1) Коммить код
2) Писать хронику экспериментов в wiki
3) Писать в инструкциях, как запустить обучение



Разработка модели



	Было	Стало
Собрать данные		
Обучить модель		
Провалидировать модель		
Поместить модель в место, доступное для прода		

Разработка модели



	Было	Стало
Собрать данные	**Где-то сохраняем данные**	
Обучить модель		
Провалидировать модель		
Поместить модель в место, доступное для прода		

Data Registry



Версионируем датасеты с помощью DVC, сохраняем их на hdfs.

Используем для обучения только данные доступные на продакшне.



Разработка модели



	Было	Стало
Собрать данные	Где-то сохраняем данные	**Регистрируем данные в Data Registry**
Обучить модель		
Провалидировать модель		
Поместить модель в место, доступное для прода		

Разработка модели



	Было	Стало
Собрать данные	Где-то сохраняем данные	Регистрируем данные в Data Registry
Обучить модель	**Код в отдельных скриптах или ноутбуках**	
Провалидировать модель		
Поместить модель в место, доступное для прода		

Jupyter Notebook



jupyter BOT CLASSIFICATION MODEL -- BEST ATTEMPT Last Checkpoint: 8 минут назад (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

In [1]:
```
import data_processing
```

In [2]:
```
import pandas as pd
```

In [3]:
```
data = pd.read_csv("data/behaviour_public_2018-2019_filtered_raw.csv", "\t")
features = data_processing.generate_features(data)
target = data_processing.get_target(data)
model = BotClassificationPipeline()
model.fit(features, target)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-3-c940d1691b65> in <module>
      2 features = data_processing.generate_features(data)
      3 target = data_processing.get_target(data)
----> 4 model = BotClassificationPipeline()
      5 model.fit(features, target)

NameError: name 'BotClassificationPipeline' is not defined
```

Инструменты для пайплайнов





DVC



DAGSTER

Задачи, которые мы решаем



Спам или нет

Матричная факторизация для музыки

Есть ли на картинке текст?

Порно VS Эротика VS Норм

Кликбейт или нет

Какая тематика у поста

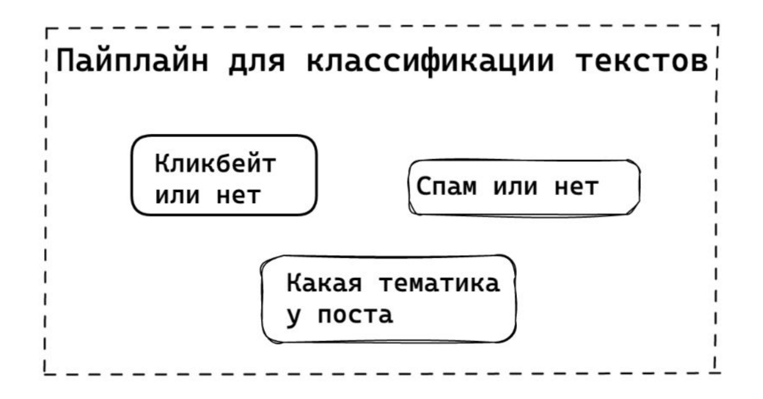
Что изображено на фото?

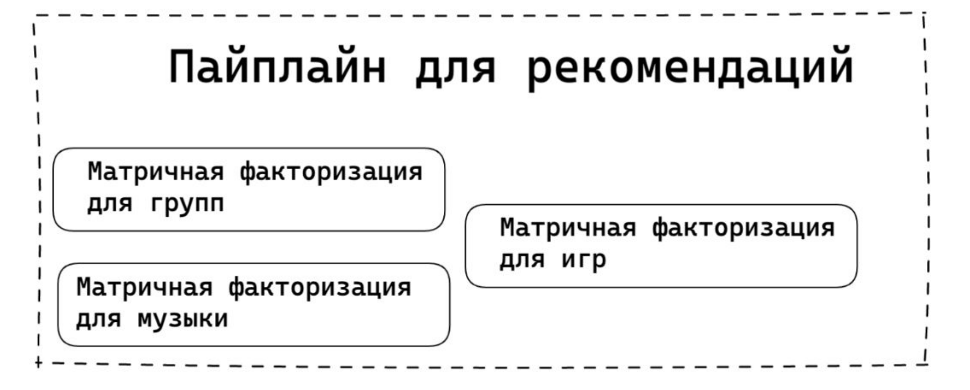
Матричная факторизация для групп

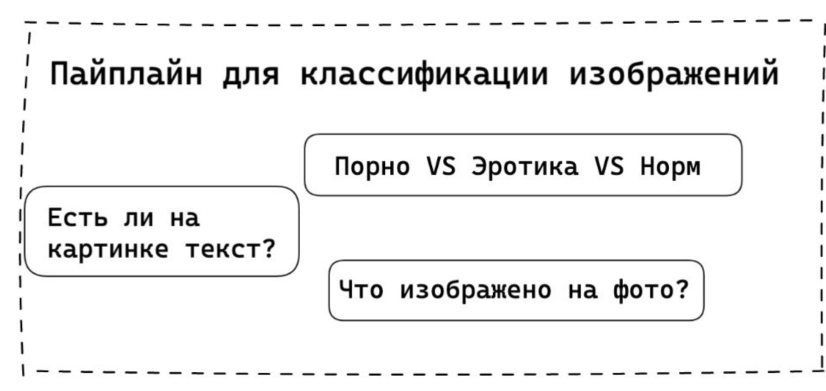
Матричная факторизация для игр

Задачи, которые мы решаем — похожи друг на друга!









Работа с кодом



Датасет

Параметры

Пайплайн
для рекомендаций

Модель для рекомендаций музыки

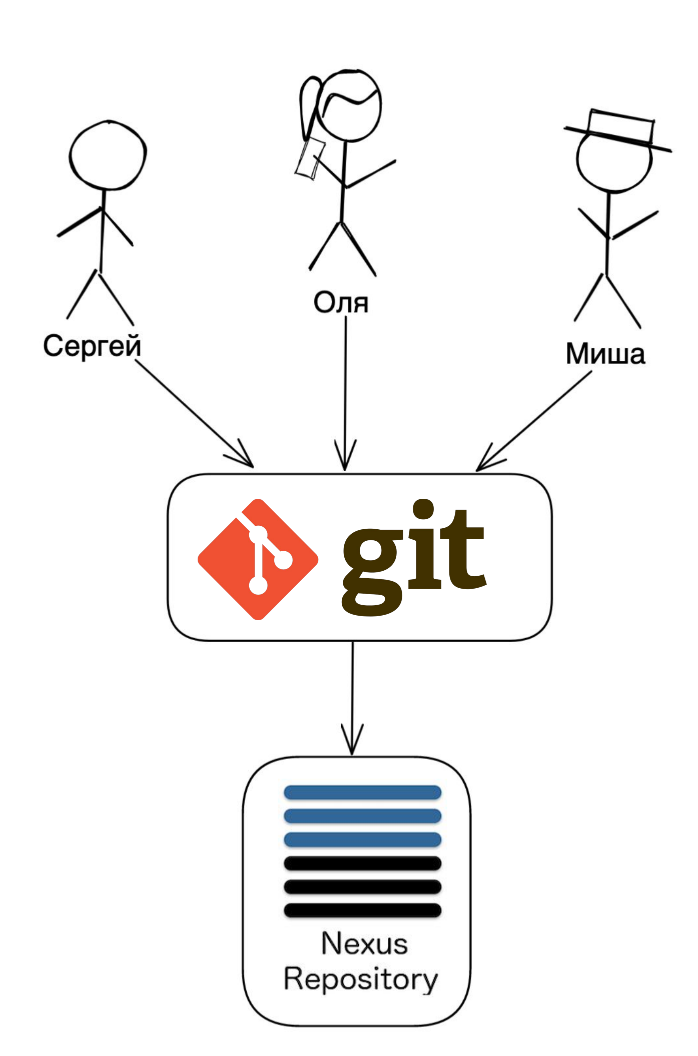
Модель для рекомендаций групп

Модель для рекомендаций
в discovery

Работа с кодом

- Обязательное ревью кода (уклон на переиспользование наработок)

- Покрытия тестами ML-компонентов



Разработка модели



	Было	Стало
Собрать данные	Где-то сохраняем данные	Регистрируем данные в Data Registry
Обучить модель	**Код в отдельных скриптах или ноутбуках**	**Переиспользуемые пайплайны**
Провалидировать модель		
Поместить модель в место, доступное для прода		

Разработка модели



	Было	Стало
Собрать данные	Где-то сохраняем данные	Регистрируем данные в Data Registry
Обучить модель	Код в отдельных скриптах или ноутбуках, **параметры запуска в wiki**	Переиспользуемые пайплайны
Провалидировать модель		
Поместить модель в место, доступное для прода		

Работа с параметрами



Все параметры для обучения моделей

- используемые датасеты
- параметры предобработки данных
- параметры конструирования моделей
- параметры тонкости обучения

фиксируются в GIT

experiments / ok_monitoring_anomaly_detection / config / **config.yml** MODIFIED

```diff
 pipeline_type: time_series_classification_pipeline
 experiment_name: ok_monitoring_anomaly_detection

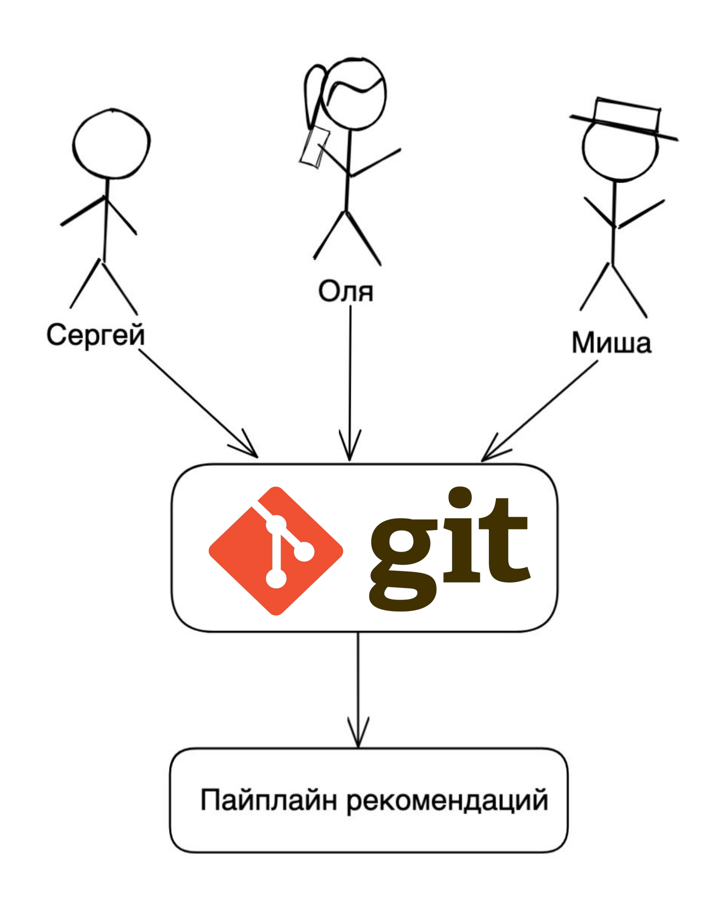
 stages:
   dataset_download:
     datasets:
-      - {dataset_name: ok_monitoring_time_series, tag: ok_
-        out: data_20_08}
-      - {dataset_name: ok_monitoring_time_series, tag: ok_
-        out: &train_folder data_20_09}
-      - {dataset_name: ok_monitoring_time_series, tag: ok_
-         out: &val_folder data_20_10}
+      - {dataset_name: ok_monitoring_time_series, tag: ok_
+      - {dataset_name: ok_monitoring_time_series, tag: ok_
+      - {dataset_name: ok_monitoring_time_series, tag: ok_

   concat_datasets:
     inputs:
       - type: csv
         paths:
-          - data_20_08/data.csv
           - data_20_09/data.csv
-        out: data_20_09/concated_data.csv
+          - data_20_10/data.csv
+        out: data_20_10/concated_data.csv

   split_dataset:
     inputs:
       - test_size: 0.0
         val_size: 0.5
         train_size: 0.5
-        input: data_20_10/data.csv
-        val_path: data_20_10/val.csv
-        train_path: data_20_10/test.csv
+        input: data_20_11/data.csv
+        val_path: data_20_11/val.csv
```

Работа с параметрами





Разработка модели



	Было	Стало
Собрать данные	Где-то сохраняем данные	Регистрируем данные в Data Registry
Обучить модель	Код в отдельных скриптах или ноутбуках, **параметры запуска в wiki**	Переиспользуемые пайплайны, **параметры фиксируются в GIT**
Провалидировать модель		
Поместить модель в место, доступное для прода		

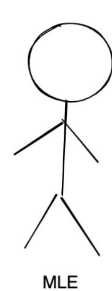
Разработка модели



	Было	Стало
Собрать данные	Где-то сохраняем данные	Регистрируем данные в Data Registry
Обучить модель	Код в отдельных скриптах или ноутбуках, параметры запуска в wiki **Никто не контролирует процесс обучения**	Переиспользуемые пайплайны, параметры фиксируются в GIT
Провалидировать модель		
Поместить модель в место, доступное для прода		

Человек запускает обучение





Датасет

Параметры

Пайплайн
для рекомендаций

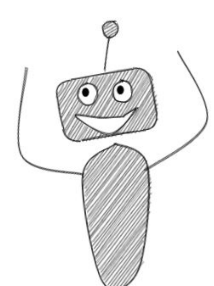
Модель для рекомендаций музыки

Модель для рекомендаций групп

Модель для рекомендаций
в discovery

Робот запускает обучение





Датасет

Параметры

Пайплайн
для рекомендаций

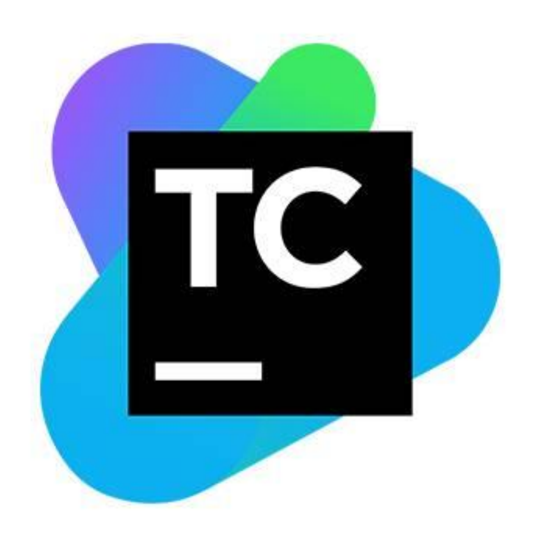
Модель для рекомендаций музыки

Модель для рекомендаций групп

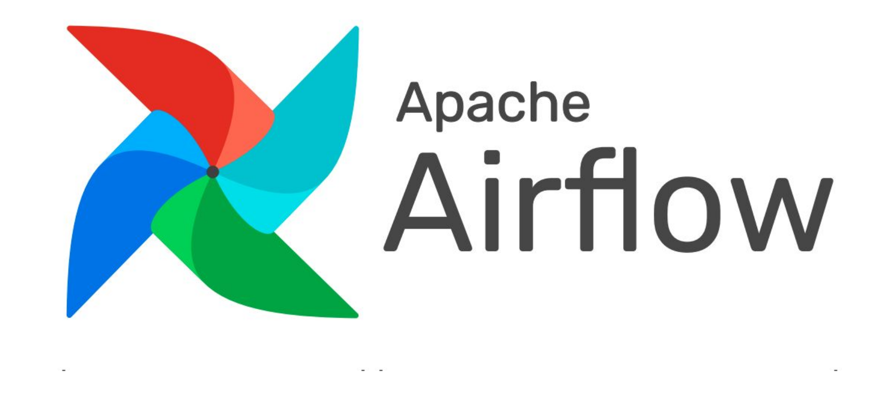
Модель для рекомендаций
в discovery

Механизмы запуска





Коммитим параметры, по ним обучается модель (CI/CD для модели)



Строим большие пайплайны со Spark и переобучением

Разработка модели



	Было	Стало
Собрать данные	Где-то сохраняем данные	Регистрируем данные в Data Registry
Обучить модель	Код в отдельных скриптах или ноутбуках, параметры запуска в wiki **Никто не контролирует процесс обучения**	Переиспользуемые пайплайны, параметры фиксируются в GIT **Обучение происходит в контролируемой среде**
Провалидировать модель		
Поместить модель в место, доступное для прода		

Разработка модели



	Было	Стало
Собрать данные	Где-то сохраняем данные	Регистрируем данные в Data Registry
Обучить модель	Код в отдельных скриптах или ноутбуках, параметры запуска в wiki Никто не контролирует процесс обучения	Переиспользуемые пайплайны, параметры фиксируются в GIT Обучение происходит в контролируемой среде
Провалидировать модель	**Отчет с метриками делается в jupyter notebook или html и прикрепляется в wiki/git**	
Поместить модель в место, доступное для прода		

MLFLOW TRACKING



Логирование метрик и ключевых параметров

Experiments

Search Experiments

Default

Default

Experiment ID : 0

Artifact Location :
file:///Users/mikhail.maryufich/PycharmProjects/mlflow_tutorial/examples/sklearn_elasticnet_wine/mlruns/0

Notes

None

Search Runs: metrics.rmse < 1 and params.model = "tree" and tags.mlflow.source.type = "LOCAL" State: Active Search Clear

Showing 4 matching runs Compare Delete Download CSV Columns

86					Parameters		Metrics		
Start Time	Run Name	User	Source	Version	alpha	l1_ratio	mae	r2	rmse
2020-10-21 20:11:55	-	mikhail.mar...	train.py	-	0.1	0.5	0.562	0.243	0.731
2020-10-21 20:11:51	-	mikhail.mar...	train.py	-	0.1	0.4	0.56	0.245	0.73
2020-10-21 20:11:47	-	mikhail.mar...	train.py	-	0.4	0.4	0.597	0.172	0.764
2020-10-21 20:11:43	-	mikhail.mar...	train.py	-	0.4	0.3	0.586	0.193	0.755

MLFLOW TRACKING



Parameters

Name	Value
elasticNetParam	1.0
regParam	0.08

Metrics

Name	Value
auc on test	0.734
auc on train	0.731
auc_pr on test	0.396
auc_pr on train	0.398

MLFLOW TRACKING



Есть возможность сохранять артефакты

▾ Artifacts

- ▾ workdir
  - ▸ model_config
  - ▸ trace_serialization
  - 20200624_164935.log
  - best.pth
  - epoch_27.pth
  - epoch_29.pth
  - epoch_30.pth
- config.yml

Full Path: hdfs://datalab-hadoop-nn/var/artifacts/mlflow/214/8146454960bd4321a2c8021f5cf44eb8/artifacts/workdir/20...
Size: 3KB

```
2020-06-24 17:08:43,475 - INFO - Best checkpoint was changed: epoch_1.pth with 0.910159144363855 from -i
2020-06-24 17:29:32,759 - INFO - Best checkpoint was changed: epoch_2.pth with 0.9258855909499958 from 0
2020-06-24 17:47:26,562 - INFO - Best checkpoint was changed: epoch_3.pth with 0.9345885924071096 from
0.9258855909499958
2020-06-24 18:08:40,234 - INFO - Best checkpoint was changed: epoch_4.pth with 0.9390506371820738 from
0.9345885924071096
2020-06-24 18:31:57,720 - INFO - Best checkpoint was changed: epoch_5.pth with 0.9411203934084562 from
0.9390506371820738
2020-06-24 18:57:15,263 - INFO - Best checkpoint was changed: epoch_6.pth with 0.9426317428210148 from
```

Работа с метриками



Датасет

Параметры

Пайплайн
для рекомендаций

Модель для рекомендаций музыки

Модель для рекомендаций групп

Модель для рекомендаций
в discovery

mlflow

Разработка модели



	Было	Стало
Собрать данные	Где-то сохраняем данные	Регистрируем данные в Data Registry
Обучить модель	Код в отдельных скриптах или ноутбуках, параметры запуска в wiki Никто не контролирует процесс обучения	Переиспользуемые пайплайны, параметры фиксируются в GIT Обучение происходит в контролируемой среде
Провалидировать модель	**Отчет с метриками делается в jupyter notebook или html и прикрепляется в wiki/git**	**Метрики и отчеты сохраняются в mlflow с привязкой к запуску**
Поместить модель в место, доступное для прода		

Разработка модели



	Было	Стало
Собрать данные	Где-то сохраняем данные	Регистрируем данные в Data Registry
Обучить модель	Код в отдельных скриптах или ноутбуках, параметры запуска в wiki Никто не контролирует процесс обучения	Переиспользуемые пайплайны, параметры фиксируются в GIT Обучение происходит в контролируемой среде
Провалидировать модель	Отчет с метриками делается в jupyter notebook или html и прикрепляется в wiki/git	Метрики и отчеты сохраняются в mlflow с привязкой к запуску
Поместить модель в место, доступное для прода	**Просто создаем папочку в HDFS, версионирование названиями**	

MLFLOW MODEL REGISTRY



mlflow GitHub Docs

Default > Run ed089cf2d5d24b8880446761987b7ac2

Date: 2019-10-16 22:49:25

Source: train_predict.py

Git Commit: fcfefecd6f830c43dc8cdc048b83ac92fd8d5d7e

User: sid.murching

Duration: 6.2s

Notes

None

Parameters

Metrics

Tags

Name	Value	Actions
	No tags found.	

Add Tag

Name Value Add

Artifacts

- model
  - MLmodel
  - conda.yaml
  - tfmodel

Full Path: /tmp/swag/0/ed089cf2d5d24b8880446761987b7ac2/artifacts/model

Size: 0B

Register Model

MLFLOW MODEL REGISTRY



mlflow GitHub Docs

Default > Run ed089cf2d5d24b8880446761987b7ac2

Date: 2019-10-16 22:49:25

Source: train_predict.py

Git Commit: fcfefecd6f830c43dc8cdc048b83ac92fd8d5d7e

User: sid.murching

Duration: 6.2s

Notes

None

Parameters

Metrics

Tags

Name	Va
	No

Add Tag

Name | Value | Add

Register Model

Once registered, the model will be available in the model registry and become public.

* Model

+ Create New Model

* Model Name

Model A

Cancel | Register

Artifacts

- model
  - MLmodel
  - conda.yaml
  - tfmodel

Full Path: /tmp/swag/0/ed089cf2d5d24b8880446761987b7ac2/artifacts/model

Size: 0B

Register Model

MLFLOW MODEL REGISTRY



mlflow GitHub Docs

Registered Models

search model name

Name	Latest Version	Staging	Production	Last Modified
Model A	Version 1	Version 1	–	2019-10-16 22:51:19
Model B	Version 1	–	–	2019-10-16 22:51:52

94

1

MLFLOW MODEL REGISTRY



Ко всему этому, есть rest api и возможность делать кодом из клиента
Это позволяет строить разные релизные циклы

```
client = MlflowClient()
result = client.create_model_version(
    name="sk-learn-random-forest-reg-model",
    source="mlruns/0/d16076a3ec534311817565e6527539c0/artifacts/sklearn-model",
    run_id="d16076a3ec534311817565e6527539c0"
)
```

95

Схема с Model Registry



Датасет

Параметры

Пайплайн
для рекомендаций

mlflow

mlflow
Model Registry

Модель для рекомендаций музыки

Модель для рекомендаций групп

Модель для рекомендаций
в discovery

Разработка модели



	Было	Стало
Собрать данные	Где-то сохраняем данные	Регистрируем данные в Data Registry
Обучить модель	Код в отдельных скриптах или ноутбуках, параметры запуска в wiki Никто не контролирует процесс обучения	Переиспользуемые пайплайны, параметры фиксируются в GIT Обучение происходит в контролируемой среде
Провалидировать модель	Отчет с метриками делается в jupyter notebook или html и прикрепляется в wiki/git	Метрики и отчеты сохраняются в MLflow с привязкой к запуску
Поместить модель в место, доступное для прода	**Просто создаем папочку в HDFS, версионирование названиями**	**Модель сохраняется и версионируются в MLflow Model Registry**

Итоговая схема



Jira

Улучшить систему рекомендаций

MLE

git

Артефакты труда

Код

Параметры

Данные

USER

Storage

Использование модели

Apache Airflow

Pipeline

Apache Airflow

Собрать данные

Обучить модель

Провалидировать модель

Поместить модель в место, доступное для прода

mlflow Model Registry

Результаты



- Научились воспроизводить обучение моделей
- Time2Market для типовых моделей снижен до 2-х дней
- Большинство моделей в ОК прошли через описанные процедуры

Организационные решения



- На продакшн не может попасть модель, которая не прошла через автоматическое обучение с фиксацией кода/параметров/данных.

- Внимание качеству кода и тестов уделяется не меньше, чем исследовательской составляющей

Доклад про воспроизводимость



SmartData

Михаил Марюфич
Mail.Ru Group

CI/CD для Ml-моделей и датасетов

Машинное обучение в продакшне — это просто! Нужно только...



- Автоматизация-автоматизация-автоматизация!
- Технический процесс так же важен, как и счастье пользователей!
- Результат работы DS-специалиста не модель, а пайплайн!



Итоги



Сегодня мы пришли к :

- автоматизации batch inference
- мониторингу данных
- автоматизации обучения модели





О чем не поговорили



- Model as a service
- Streaming



Машинное обучение в продакшне — это просто! Нужно только....

Михаил Марюфич



